近似稀疏的低秩张量填充*

喻高航†, 郑伟东‡, 胡文玉§¶

摘要: 针对目前大多数的低秩张量填充(LRTC)模型存在过度稀疏而导致数据的细微特征被忽略的现象, 本文借助框架变换和低秩矩阵分解, 提出了一个基于近似稀疏的低秩张量填充(AS-LRTC) 模型, 进一步设计了块逐次上界极小化(BSUM) 算法求解该模型. 在一定条件下可以证明该算法的收敛性, 大量的实验结果表明本文提出的算法比现有一些经典算法有明显的优势.

关键词: 近似稀疏, 低秩, 张量填充, 矩阵分解, 框架变换

1 引言

大数据时代每时每刻都在产生大量的高维数据, 例如, 电商交易数据、多传感器阵列数据、疾病基因表达数据、网络社交数据、视频监控、临床医学影像等. 这些数据不仅规模大, 维度高, 还有着较为复杂的内在结构. 若采用矩阵对数据进行表示, 将会损坏数据的各维度的信息结构而造成较大的误差. 张量(tensor)作为向量和矩阵的高维推广, 是高维数据的一种自然表达形式. 采用张量形式表示和处理高维数据已经在许多领域得到了广泛的应用, 如图像/视频修复[1, 2]、前景后景分离[3]、核磁共振图像恢复[4]、高光谱图像恢复[5, 6]、视频去雨[7, 8] 以及信号重建[9] 等, 并且获得了很好的效果.

张量填充是指从部分的观测数据中恢复完整的数据, 可以被描述为一个张量数据缺失值估计问题(即填充问题), 是矩阵填充的高阶扩展. 问题的关键是建立观测数据和缺失数据之间的联系[10]. 低秩张量填充(Low-Rank Tensor Completion, 简称LRTC)是利用真实张量的低秩先验来估计缺失数据, 从数学上讲, 低秩张量填充模型写成

$$\begin{aligned} \min_{\mathcal{Y}} \ & rank(\mathcal{Y}) \\ \text{s.t.} \ & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \tag{1}$$

其中$\mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ 是待恢复张量, $\mathcal{F}$ 是观测张量, Ω 是记录观测数据对应位置组成的集合, $\mathcal{P}(\cdot)$ 是投影算子. 张量的秩有多种不同的定义且不唯一, 在张量秩的所有定义中, Tucker 秩被广泛用于低秩张量填充问题研究中[10–12]. 然而, 模型(1) 中直接使用Tucker 秩进行求解是一个NP 难问题[13].

矩阵核范数(也称为迹范数或Ky Fan范数)是秩函数在矩阵谱范数单位球上凸包络, 因此, 通常利用核范数来逼近矩阵秩函数. 受此启发, Liu 等人[10] 首次提出了张量核范数的定义, 即张量每个模矩阵的核

*国家自然科学基金(批准号: 11661007,61863001), 浙江省自然科学基金重大项目(批准号: LD19A010002)和江西省自然科学基金(批准号: 20181BAB202021,20192BAB205086) 资助项目

†杭州电子科技大学理学院, 杭州 310018
‡赣南师范大学数学与计算机科学学院, 赣州 341000
§赣南师范大学数学与计算机科学学院, 赣州 341000
¶通讯作者. E-mail: huwenyu@gnnu.edu.cn

范数加权和:

$$\| \mathcal{Y} \|_* = \sum_{n=1}^{N} \alpha_n \|Y_{(n)}\|_*. \tag{2}$$

其中$\alpha_n > 0$ 且$\sum_{n=1}^{N} \alpha_n = 1$, $Y_{(n)}$表示张量$\mathcal{Y}$沿第n模式展开的矩阵, $\|\cdot\|_*$ 表示矩阵核范数. 基于张量核范数的定义, 他们提出了如下张量填充模型:

$$\min_{\mathcal{Y}} \sum_{n=1}^{N} \alpha_n \|Y_{(n)}\|_*$$
$$\text{s.t.} \ \ \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}.$$

之后, 很多有效的算法被提出来求解上述凸优化问题. 例如, 文献[10]设计了三种求解算法(SiLRTC、FaLRTC 和HaLRTC), 而Gandy等人[14] 提出了基于Douglas-Rachford 分裂算法和交替方向乘子法(ADMM)的求解算法.

然而, Romera-Paredes 等人[15] 证明了张量每个模矩阵的核范数加权和并不是张量展开矩阵秩的和的最紧凑凸包. 于是, 有学者考虑非凸优化模型, 如Li 等人[16] 提出了基于矩阵p-范数的非凸松弛模型:

$$\min_{\mathcal{Y}} \sum_{n=1}^{N} \alpha_n \|Y_{(n)}\|_{p,\varepsilon}^p$$
$$\text{s.t.} \ \ \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F},$$

其中$\varepsilon > 0, p \in (0,1)$. 而Ji等人[17] 提出了基于每个模式展开的非凸对数函数模型:

$$\min_{\mathcal{Y}} \sum_{n=1}^{N} \alpha_n \text{logdet}((Y_{(n)} Y_{(n)}^T)^{1/2} + \varepsilon_n I_n)$$
$$\text{s.t.} \ \ \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F},$$

其中$\varepsilon_n > 0$. 文献[17] 将对数函数进行线性化处理, 应用了ADMM 方法求解模型, 但未分析算法的收敛性.

Mu等人[18] 证明了张量每个模核范数的和实质上是次优的, 并提出了新的凸松弛方法, 即方处理(Square Deal). 相对于张量核范数, 方处理要求更少的观察样本可以得到可靠的恢复结果. 进一步, Nie[19] 证明了在一定条件下, 张量方处理所得矩阵的秩可以与张量的CP 秩相等. 基于方处理, Bai 等人[20] 提出了一个修正的约束张量填充模型, 并建立了其误差边界.

$$\min_{\mathcal{Y}} \sum_{n=1}^{N} \| \mathcal{Y}_{\lfloor j \rfloor} \|_* - \langle F(\tilde{\mathcal{Y}}_{\lfloor j \rfloor}), \mathcal{Y}_{\lfloor j \rfloor} - \tilde{\mathcal{Y}}_{\lfloor j \rfloor} \rangle$$
$$\text{s.t.} \ \ \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \ \| \mathcal{Y} \|_\infty \leq c,$$

其中$\tilde{\mathcal{Y}} \in R^{n_{i_1} \times \cdots \times n_{i_N}}$ 一个张量, 通过将$\mathcal{Y}$ 的模-i_k 重标记为模-$k(k = 1, 2, \cdots, N)$, $j \in \{1, 2, \cdots, N\}$ 和置换$(i_1, \cdots, i_n)$, 使$n_{(1)} := \prod_{k=1}^{j} n_{i_k}$ 尽可能接近$n_{(2)} := \prod_{k=j+1}^{N} n_{i_k}$, $\mathcal{Y}_{\lfloor j \rfloor} = reshape(\tilde{\mathcal{Y}}_{(1)}, n_{(1)}, n_{(2)})$.

上述介绍的方法在求解过程中需要进行矩阵的奇异值分解(SVD), 随着维数的增加, 这在时间和存储上都是高消耗. Xu 等人[21] 将低秩矩阵分解的矩阵填充模型[22] 推广到低秩张量填充模型, 通过并行地对张量$\mathcal{Y}$ 沿所有模式展开的矩阵应用低秩矩阵分解:

$$\min_{\mathcal{Y}, A, X} \sum_{n=1}^{N} \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2$$
$$\text{s.t.} \ \ \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \tag{3}$$

其中$Y_{(n)} \approx A_n X_n$, $A = (A_1, \cdots, A_N)$, $X = (X_1, \cdots, X_N)$. 该方法称为TMac[21], 获得比FaLRTC 更快的时间效率和更好的性能.

实际数据不仅限于其低秩特性, 往往还具有分段光滑的性质. 在模型(3) 中, X_3 在实际应用中具有清晰的物理意义, 例如: 视频数据表现为数据的空间信息结构, 常常表现出分段光滑的性质. 特别地, 全变分(Total Variation,TV)能够很好地刻画分段光滑性质, 并且在很多应用中取得了显著的效果[4, 23–25]. 针对实际应用中X_3 表现出的分段光滑的特性, Ji 等人[26] 将TV 正则项与低秩张量填充相结合, 提出如下模型:

$$\begin{aligned} &\min_{\mathcal{Y},X,A} \sum_{n=1}^{3} \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \mu \mathrm{TV}(X_3) \\ &\text{s.t.} \;\; \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \end{aligned} \tag{4}$$

其中$\mu > 0$ 是正则参数, $\sum_{n=1}^{3} \alpha_n = 1$, $A = (A_1, A_2, A_3)$, $X = (X_1, X_2, X_3)$, TV(X_3)表示X_3 的全变分. 由于TV正则项通常会导致阶梯伪影, 因子图像的细节和几何特征在模型(4) 中不能很好的保留.

在实际应用中, 细节和几何结构具有重要的地位. 特别地, 框架变换是一组正交基的生成, 放宽了正交性和线性无关性的要求, 它的冗余具有保留几何特征和细节的能力. 基于此, Jiang 等人[27] 将框架变换正则项引入到张量填充问题中, 得到优于模型(4) 模的效果, 其模型如下:

$$\begin{aligned} &\min_{\mathcal{Y},X,A} \sum_{n=1}^{3} \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \lambda \parallel W X_3^T \parallel_{1,1} \\ &\text{s.t} \;\; \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \end{aligned} \tag{5}$$

其中$\lambda > 0$ 是正则参数, W 是框架变换矩阵, 满足$W^T W = I$. 模型(4) 和模型(5) 的目标函数不是关于变量$(X, A, \mathcal{Y})$ 的联合凸函数, 但是关于变量$X, A, \mathcal{Y}$ 的子函数是凸的. 因此, 采用块逐次上界极小化算法进行求解, 且具有收敛性保证.

然而，图像经过框架变换后的系数向量通常是近似稀疏而非完全稀疏, 在模型(5) 中采用l_1范数会导致恢复的因子图像X_3 的微妙特征被忽略掉. 因此, 在图像修复问题中, Shen等人[28] 采用框架变换的软阈值算子, 以及对l_0 范数采用加权l_1 范数进行近似, 最后建立了近似稀疏的修复模型, 得到了优于l_1范数正则化的修复模型. 受此启发, 本文对模型(5) 进行改进, 提出一个近似稀疏的低秩张量填充模型. 在此基础上，基于块逐次上界极小化算法(Block Successive Upper-bound Minimization, 简称BSUM)设计了快速有效的优化算法, 并给出了算法的收敛性证明. 同时, 通过数值实验来说明本文算法的优越性.

本文内容结构安排如下: 第二章回顾了有关预备知识, 如张量的基础知识、框架变换、不动点算法和BSUM 算法的一些初步知识等. 第三章给出了一个近似稀疏张量填充模型, 并基于BSUM算法与临近不动点算法, 并分析了算法的收敛性; 第四章给出了数值实验; 最后一章总结全文.

2 张量基础与预备知识

本章首先将介绍相关记号及涉及到的有关张量基础知识[29], 同时还将简要介绍本文用到的相关算法和框架变换理论.

2.1 相关记号

本文采用希腊字母表示标量, 例如: α, β, μ; 粗体小写字母表示向量, 例如: $\mathbf{a}, \mathbf{b}$; 大写字母表示矩阵,

例如: **X**, **Y**; 花体字母表示张量, 例如: $\mathcal{X},\mathcal{Y}$; 空心字母表示集合, 例如: $\mathbb{R}$ 表示实数空间. 特别地, Ω 表示观测数据中已知元素的位置, 即若$\Omega_{i_1,\cdots,i_N}=1$, 则表示数据中第$(i_1,\cdots,i_N)$ 个元素已知; 若$\Omega_{i_1,\cdots,i_N}=0$, 则表示相应位置元素缺失, 张量$\mathcal{X}$ 的第$(i_1,\cdots,i_N)$ 个元素记为$x_{i_1,\cdots,i_N}$.

集合$\mathbb{S}$ 的指示函数$\iota_{\mathbb{S}}$, 定义为:

$$\iota_{\mathbb{S}}(x)=\begin{cases}0, & x\in\mathbb{S};\\ \infty, & \text{其它}.\end{cases}$$

指标张量Ω 上的投影算子记为$\mathcal{P}_\Omega$, 定义为:

$$(P_\Omega(\mathcal{X}))_{i_1,\cdots,i_N}=\begin{cases}x_{i_1,\cdots,i_N}, & \Omega_{i_1,\cdots,i_N}=1;\\ 0, & \text{其它}.\end{cases}$$

2.2 张量的定义与运算

张量(tensor) 定义为一个多维数组, 是向量和矩阵在多维空间中的推广, 数据的维度称为张量的阶. 下面我们介绍张量中涉及的一些概念以及张量与矩阵的相互转化关系.

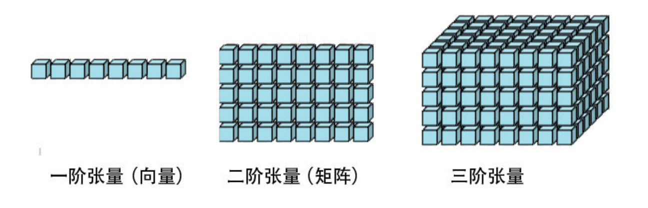


图 1: 一阶、二阶、三阶张量

定义 2.1 (张量纤维(fiber)) 纤维是指从张量中抽取向量的操作. 张量纤维是由矩阵行、列概念的高维推广. 在矩阵中固定其中一个维度, 可以得到行或者列. 类似于矩阵操作, 张量第n 维度纤维*(mode-n fiber)* 定义为固定张量其它维度指标不变, 只保留张量第n 维度指标变化生成的向量.

特别地, 矩阵的模-1纤维是矩阵的列, 模-2纤维是矩阵的行, 三维张量的模-1、-2、-3纤维分别称为列(column)、行(row) 和管(tube).

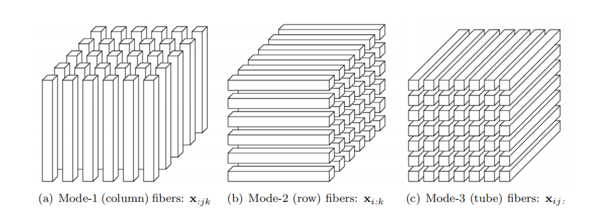


图 2: 三阶张量的模-n纤维

定义 2.2 (张量切片(slice)) 切片操作是指在张量中抽取矩阵的操作. 在张量中如果保留两个维度变化, 其它的维度不变可以得到一个矩阵, 这个矩阵即为张量切片.

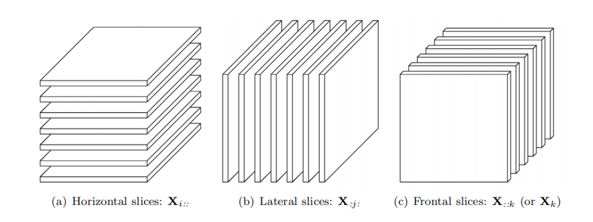


图 3: 三阶张量的切片

特别地, 三阶张量$\mathcal{X} \in R^{I_1 \times I_2 \times I_3}$ 三个维度的切片分别为$X_{i::}, X_{:j:}, X_{::k}$, 分别记为水平切片(horizontal slice)、侧向切片(lateral slice)、正向切片(frontal slice).

定义 2.3 (张量矩阵展开(Unfolding-Matricization)) *张量$\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \cdots \times d_N}$的矩阵展开是将一个张量的元素重新排列(即对张量的模$-n$ 纤维按字典顺序重新排列), 得到一个矩阵的过程. 张量在第n维度上的展开矩阵表示为$X_{(n)} = unfold_n(\mathcal{X}) \in R^{d_n \times \prod_{i \neq n} d_i}$. 其中张量的第$(i_1, \cdots, i_N)$ 个元素映射到展开矩阵的第(i_n, j) 个元素, 满足*

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k, J_k = \prod_{m=1, m \neq n}^{k-1} d_m.$$

定义 2.4 (张量内积(Inner product)) *两个相同大小张量的内积$\langle \cdot, \cdot \rangle$ 定义为它们对应元素的乘积之和, 即*

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1, i_2 \cdots, i_N} x_{i_1, i_2 \cdots, i_N} y_{i_1, i_2 \cdots, i_N}.$$

特别地, 张量的Frobenius 范数与矩阵的Frobenius 范数相同, 即所有元素的平方和开方: $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1, i_2 \cdots, i_N} |x_{i_1, i_2 \cdots, i_N}|^2}$.

定义 2.5 (张量Tucker 秩) *张量$\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \cdots \times d_N}$ 的第n维展开矩阵的秩定义为:*

$$rank_n(\mathcal{X}) = rank(X_{(n)}).$$

令$rank_n(\mathcal{X}) = r_n, n = 1, 2, \cdots, N$, 则张量$\mathcal{X}$ 的Tucker 秩定义为:

$$rank_t(\mathcal{X}) = (r_1, r_2, \cdots, r_N). \tag{6}$$

2.3 BSUM算法

假设$X = X_1 \times X_2 \times \cdots \times X_n$且$X_i \subset \mathbb{R}^{m_i}$, 变量$\mathbf{x} \in X$可视为$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$, 其中第$i$分块变量$\mathbf{x}_i \in X_i$. 考虑如下的优化问题:

$$\begin{aligned} &min \quad f(\mathbf{x}) \\ &s.t \quad \mathbf{x} \in X. \end{aligned} \tag{7}$$

令$u_i(\mathbf{x}_i, \mathbf{x}^k) = f(\mathbf{x}_1^k, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_n^k) + \frac{\rho}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_2^2$, 其中$\rho > 0, i = 1, \cdots, n$. BSUM算法每一次更新

单个变量块, 不断循环进行如下n步迭代, 直到满足一定的收敛条件.

$$\begin{cases} step\ 1: \mathbf{x}_1^{k+1} = \underset{\mathbf{x}_1 \in \mathbf{X}_1}{argmin}\ u_1(\mathbf{x}_1, \mathbf{x}_{(1)}^k), \\ step\ 2: \mathbf{x}_2^{k+1} = \underset{\mathbf{x}_2 \in \mathbf{X}_2}{argmin}\ u_2(\mathbf{x}_2, \mathbf{x}_{(2)}^k), \\ \qquad \vdots \\ step\ n: \mathbf{x}_n^{k+1} = \underset{\mathbf{x}_n \in \mathbf{X}_n}{argmin}\ u_n(\mathbf{x}_n, \mathbf{x}_{(n)}^k), \end{cases} \tag{8}$$

其中$\mathbf{x}_{(r)}^k = (\mathbf{x}_1^{k+1}, \cdots, \mathbf{x}_{r-1}^{k+1}, \mathbf{x}_r^k \cdots, \mathbf{x}_n^k), r = 1, \cdots, n, k = 1, 2, \cdots$.

在满足适当条件下, 可以证明BSUM算法按坐标收敛:

引理 2.1 ([30]) *设$u(\mathbf{x}, \mathbf{x}^{k-1})$ 是$f(x)$ 在x^{k-1} 处的近似函数, 若满足以下条件:*

$$\begin{cases} u_i(\mathbf{y}_i, \mathbf{y}) = f(\mathbf{y}), \forall \mathbf{y} \in \mathcal{X}, \forall i; \\ u_i(\mathbf{y}_i, \mathbf{y}) \geq f(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{i-1}, \mathbf{x}_i, \mathbf{y}_{i+1}, \cdots, \mathbf{y}_n), \forall \mathbf{x}_i \in \mathcal{X}_i, \forall \mathbf{y} \in X, \forall i; \\ u_i^{'}(\mathbf{x}_i, \mathbf{y}; \mathbf{d}_i)|_{\mathbf{x}_i = \mathbf{y}_i} = f^{'}(\mathbf{y}; \mathbf{d}), \forall \mathbf{d} = (0, \cdots, \mathbf{d}_i, \cdots, 0)\ s.t.\ \mathbf{y}_i + \mathbf{d}_i \in X_i, \forall i; \\ u_i(\mathbf{x}_i, \mathbf{y}) \text{在} (\mathbf{x}_i, \mathbf{y}) \text{ 处是连续的}, \forall i = 1, \cdots, n, \end{cases}$$

其中$u_i(\mathbf{x}_i, \mathbf{y})$ 是关于第i个块的子问题, $f^{'}(\mathbf{y}; \mathbf{d})$ 是函数f 在$\mathbf{y}$ 处$\mathbf{d}$ 方向的方向导数. 假设$u_i(\mathbf{x}_i, \mathbf{y})$ 是关于$\mathbf{x}_i (i = 1, \cdots, n)$ 的拟凸函数, 且每个子问题$\underset{\mathbf{x}_i}{argmin}\ u_i(\mathbf{x}_i, \mathbf{x}^{k-1}), s.t.\ x \in X_i$ 在任意点$\mathbf{x}^{k-1} \in X$ 处均有唯一解, 则由BSUM 算法产生的点列按坐标收敛到问题(7)的极小值点.

2.4 框架变换(framelet变换)

如果系统$X \in L_2(\mathbb{R})$满足

$$f = \sum_{g \in X} \langle f, g \rangle g, \forall f \in L_2(\mathbb{R}),$$

则称之为$L_2(\mathbb{R})$ 的紧框架, 其中$\langle \cdot, \cdot \rangle$ 是$L_2(\mathbb{R})$ 的内积. 而小波系统$X(\Psi)$定义为由以下有限集合的扩张和仿射集合: $\Psi = \{\psi_1, \psi_2, \cdots, \psi_r\} \in L^2(R)$ 定义.

$$X(\Psi) = \{2^{k/2}\psi_l(2^k \cdot -j) : \psi_l \in \Psi; 1 \leq l \leq r; k, j, l \in Z\}.$$

特别地, 如果$X(\Psi)$ 同时是$L_2(R)$ 的紧框架, 则$\psi_l (l = 1, 2, \cdots, r)$ 被称为框架, $X(\Psi)$ 被称为紧小波框架.

在离散化情形, 图像f可等价视作$\{f_i = \langle f, \psi(\cdot - i) \rangle\}$, 其中$\psi$ 是与小框架系统相关的可细化函数. f的L级离散小框架分解则是在预定最粗糙L 层的系数$\{f = \langle f, 2^{-L/2}\psi_i(2^{-L} \cdot -j) \rangle\}$, 同时框架系数为

$$\{f = \langle f, 2^{-L/2}\psi_i(2^{-L} \cdot -j) \rangle, 1 \leq i \leq r^2 - 1, 0 \leq l \leq L.\}$$

上述分解可写成关于图像$f \in \mathcal{R}^{mn}$的线性形式Wf, 其中$W \in \mathbb{R}^{k \times mn}$. 由酉扩张原理(unitary extension principle, UEP)可知, $W^T W = I$. 其中W^T 表示框架逆变换, W 的行向量构成了$\mathbb{R}^{mn}$的一个紧框架系统. 文中采用文献[31–33] 构造的分段线性B 样条小框架.

2.5 临近不动点算法

记$\Gamma_0(\mathbb{R}^d)$ 为所有下半连续凸函数$h:\mathbb{R}^d\to(-\infty,+\infty]$ 构成的集合, 其中$dom(h):=\{\mathbf{u}\in\mathbb{R}^d: h(\mathbf{u})<+\infty\}\neq\emptyset$. 设$f,g\in\Gamma_0(\mathbb{R}^d)$, 考虑凸极小化问题:

$$\min_{\mathbf{x}\in R^n}\varphi(\mathbf{x})+\psi(\mathbf{W}\mathbf{x}). \tag{9}$$

定义 2.6 (临近算子(proximity operator)) *若f 是在$\mathbb{R}^m$ 上的实凸函数, 则函数f 在$\mathbf{x}\in\mathbb{R}^n$ 上的临近算子定义为:*

$$prox_f(\mathbf{x})=argmin\{f(\mathbf{u})+\frac{1}{2}\parallel\mathbf{u}-\mathbf{x}\parallel_2^2:\mathbf{u}\in\mathbb{R}^n\}.$$

记$\mathbb{S}_+$ 为所有对称正定矩阵集合. 设$\mathbf{H}\in\mathbb{S}_+$, $\|\mathbf{x}\|_{\mathbf{H}}:=\mathbf{x}^T\mathbf{H}\mathbf{x}$ 表示加权l_2 范数. 可以定义更一般的临近算子:

$$prox_{f,H}(\mathbf{x})=argmin\{f(\mathbf{u})+\frac{1}{2}\parallel\mathbf{u}-\mathbf{x}\parallel_H^2:\mathbf{u}\in\mathbb{R}^n\}.$$

定义 2.7 (次微分(subdifferential)) *若f 是在$\mathbb{R}^m$ 上的实凸函数, 则f 在$\mathbf{x}\in\mathbb{R}^n$ 上的次微分定义为:*

$$\partial_f(\mathbf{x})=\{\mathbf{y}\in\mathbb{R}^n:f(\mathbf{z})\geq f(\mathbf{x})+<\mathbf{y},\mathbf{z}-\mathbf{x}>,\mathbf{z}\in\mathbb{R}^n\}.$$

次微分是函数在不可微情形下的扩展, 若函数可微, 则次微分就是函数的梯度. 容易证明: $\lambda\partial f(x)=\partial_{\lambda f}(x)$, 其中$\lambda>0$.

对任意的$\mathbf{x}\in\mathbb{R}^d$. $h\in\Gamma_0(\mathbb{R}^d)$ 的共轭函数h^*定义为$h^*(y):=sup\{\langle\mathbf{x}-\mathbf{y}\rangle-h(\mathbf{x}):\mathbf{x}\in\mathbb{R}\}$. 如下定理刻画了优化问题(9) 的解.

定理 2.1 ([34, 35]) *设$\varphi\in\Gamma_0(\mathbb{R}^n)$, $\psi\in\Gamma_0(\mathbb{R}^m)$, W 为$m\times n$ 矩阵. 如果$\mathbf{x}\in\mathbb{R}^n$ 是问题(9) 的解, 那么对于任何$\lambda,\beta>0$, 都存在一个向量$\mathbf{y}\in\mathbb{R}^m$, 使得*

$$\mathbf{x}=prox_{\lambda\varphi}(\mathbf{x}-\lambda W^T\mathbf{y}); \tag{10}$$

$$\mathbf{y}=prox_{\beta\psi^*}(\mathbf{y}+\beta W\mathbf{x}). \tag{11}$$

反之, 如果存在$\lambda,\beta>0$, 则$\mathbf{x}\in\mathbb{R}^n$ 和$\mathbf{y}\in\mathbb{R}^m$ 满足方程(10) 和(11), 则$\mathbf{x}$ 是问题(9) 的一个解.

基于定理2.1, 接下来设计求解问题(9) 的算法. 对于$\mathbf{z}:=(\mathbf{x},\mathbf{y})\in\mathbb{R}^n\times\mathbb{R}^m$, 定义

$$\Psi(\mathbf{z}):=\lambda\varphi(\mathbf{x})+\beta\psi^*(\mathbf{y})$$

和

$$\mathbf{E}:=\begin{bmatrix}\mathbf{I}_n & -\lambda W^T\\ \beta W & \mathbf{I}_m\end{bmatrix}.$$

由文献[35]知,

$$prox_\Psi(\mathbf{x}):=(prox_{\lambda\varphi}(\mathbf{x}),prox_{\beta\psi^*}(\mathbf{y}))\in\mathbb{R}^n\times\mathbb{R}^m, \tag{12}$$

则方程(10) 和(11) 可转化为:

$$\mathbf{z} = prox_{\Psi}(\mathbf{E}\mathbf{z}).$$

注意到对任意非零矩阵W, $\mathbf{E}$的谱范数$\|\mathbf{E}\|_2$ 严格大于1, 简单迭代$\mathbf{z}^{k+1} = prox_{\Psi}(\mathbf{E}\mathbf{z}^k)$对于给定初值$\mathbf{z}^0$可能不收敛. 为克服矩阵$\mathbf{E}$ 的扩展性带来的困难, 将E 分裂为两个子矩阵M与$E-M$ 之和. 矩阵E的分裂可能得到收敛的迭代格式, 迭代格式如下:

$$\mathbf{x}^{k+1} = prox_{\Psi}(Mx^{k+1} + (E-M)\mathbf{x}^k). \tag{13}$$

矩阵M 的选择方法有很多, 例如若选择

$$M = \begin{bmatrix} 0 & 0 \\ 2\beta W & 0 \end{bmatrix},$$

则迭代格式(13) 变成

$$\begin{cases} \mathbf{x}^{k+1} = prox_{\lambda\varphi}(\mathbf{x}^k - \lambda W^T\mathbf{y}^k); \\ \mathbf{y}^{k+1} = prox_{\beta\psi^*}(\mathbf{y}^k + \beta W(2\mathbf{x}^{k+1} - \mathbf{x}^k)). \end{cases} \tag{14}$$

而若选择

$$M = \begin{bmatrix} 0 & -2\lambda W^T \\ 0 & 0 \end{bmatrix},$$

则变成

$$\begin{cases} \mathbf{y}^{k+1} = prox_{\lambda\varphi}(\mathbf{x}^k + \beta W\mathbf{y}^k); \\ \mathbf{x}^{k+1} = prox_{\beta\psi^*}(\mathbf{x}^k - \lambda W(2\mathbf{y}^{k+1} - \mathbf{y}^k)). \end{cases} \tag{15}$$

可以证明: 如果$\lambda\beta < \frac{1}{\|W\|_2^2}$, 则从(14) 或(15) 生成的序列$\{\mathbf{x}^{(k)}\}$ 收敛到问题(9) 的解[34, 36].

3 模型及算法分析

3.1 提出的模型

本文引入软阈值框架正则化, 考虑一个三元张量$\mathcal{Y} \in \mathbb{R}^{d_1\times d_2\times d_3}$, 建立如下的张量填充模型:

$$\begin{aligned} &\min_{\mathcal{Y},A,X} \sum_{n=1}^{3} \frac{\alpha_n}{2} \| Y_{(n)} - A_nX_n \|_F^2 + \lambda \| S_\gamma(WX_3^T) \|_0 \\ &s.t.\ P_\Omega(\mathcal{Y}) = \mathcal{F}. \end{aligned} \tag{16}$$

其中$\sum_{n=1}^{3}\alpha_n = 1, \lambda > 0$ 是正则化参数, $A := (A_1, A_2, A_3), X := (X_1, X_2, X_3)$, W是框架变换矩阵, $\|\cdot\|_0$是l_0 范数, 软阈值算子$(\mathcal{S}_\gamma(u))_i := sign(u_i)max\{| u_i | -\gamma_i, 0\}$. 可知目标函数(16) 是非凸非光滑的. 若采用l_1 范数代替l_0范数, 将非凸问题松弛为凸优化问题并不能得到很好的效果, 因为l_1范数不能像l_0范数那样能有效捕捉数据稀疏性. 因此, 为了尽可能的遵循l_0范数, 本文将l_0范数重写为具有非线性间断权函数的加权l_1范数, 然后使用连续权函数逼近不连续权函数.

对$x \in \mathbb{R}$, 定义$F : \mathbb{R} \to \mathbb{R}$,

$$F(x) := \begin{cases} \frac{1}{|x|}, & x \neq 0; \\ 0, & \text{其它}. \end{cases}$$

那么, 对于$\mathbf{u}\in\mathbb{R}^d$, 有

$$\|\mathbf{u}\|_0=\sum_{i=0}^{d}F(u_i)\mid u_i\mid,$$

即向量u 的l_0 范数可以用其第i 分量的权重$F(u_i)$ 表示为它的"加权"l_1 范数. 由于函数F在$x=0$ 时不连续, 给计算带来困难, 因此采用一系列正的连续偶函数F_ϵ 逼近不连续权函数F. 定义

$$F_\epsilon(x):=\begin{cases}\frac{1}{|x|}, & |x|>\epsilon,\\ \frac{1}{\epsilon}, & 0\le |x|\le\epsilon.\end{cases}$$

则 $\lim\limits_{\epsilon\to 0^+}F_\epsilon(x)=F(x),\ x\in\mathbb{R}\backslash\{0\}$. 此时,模型(16) 可近似写成:

$$\begin{aligned}&\min_{\mathcal{Y},A,X}\sum_{n=1}^{3}\frac{\alpha_n}{2}\parallel Y_{(n)}-A_nX_n\parallel_F^2+\lambda\sum_{i=1}^{m}F_\epsilon((\mathcal{S}_\gamma(WX_3^T))_i)\mid(\mathcal{S}_\gamma(WX_3^T))_i\mid\\ &s.t.\ P_\Omega(\mathcal{Y})=\mathcal{F}.\end{aligned}$$

也即:

$$\begin{aligned}&\min_{\mathcal{Y},A,X}\sum_{n=1}^{3}\frac{\alpha_n}{2}\parallel Y_{(n)}-A_nX_n\parallel_F^2+\lambda\parallel\Xi\mathcal{S}_\gamma(WX_3^T)\parallel_1\\ &s.t.\ P_\Omega(\mathcal{Y})=\mathcal{F},\end{aligned}$$

其中Ξ是具有正对角元素的对角矩阵, Ξ 的第i 个对角线元素是$\xi_i:=F_\epsilon((\mathcal{S}_\gamma(WX_3^T)))_i$. 因此, 优化问题(16) 可以转化为如下问题求解:

$$\min\ f(\mathcal{Y},A,X)=\sum_{n=1}^{3}\frac{\alpha_n}{2}\|Y_{(n)}-A_nX_n\|_F^2+\lambda\parallel\Xi\mathcal{S}_\gamma(WX_3^T)\parallel_1+l(\mathcal{Y}),\tag{17}$$

其中

$$l(\mathcal{Y}):=\begin{cases}0, & \mathcal{P}_\Omega(\mathcal{Y})=\mathcal{F},\\ \infty, & \text{其它}.\end{cases}$$

我们称式(17)为基于近似稀疏的低秩张量填充(Approximate Sparsity based Low-Rank Tensor Completion, 简称AS-LRTC)模型.

3.2 求解算法

在本节中, 我们将设计基于BSUM 算法框架的算法求解问题(17). 该问题显然不是关于变量$(X,A,\mathcal{Y})$ 的联合凸函数, 但它对于每个变量$X,A,\mathcal{Y}$ 都是凸函数. 设$\mathcal{Z}=(X,A,\mathcal{Y})$, $Z^k=(X^k,A^k,\mathcal{Y}^k)$, 问题(17) 可以通过如下问题求解:

$$\mathcal{Z}^{k+1}=arg\ \min_{\mathcal{Z}}\ h(\mathcal{Z},\mathcal{Z}^k)=arg\ \min_{\mathcal{Z}}\ f(\mathcal{Z})+\frac{\rho}{2}\parallel\mathcal{Z}-\mathcal{Z}^k\parallel_F^2,\tag{18}$$

其中$\rho>0$. 令$\mathcal{Z}_1^k=(X^k,A^k,\mathcal{Y}^k)$, $\mathcal{Z}_2^k=(X^{k+1},A^k,\mathcal{Y}^k)$, $\mathcal{Z}_3^k=(X^{k+1},A^{k+1},\mathcal{Y}^k)$, 原问题可以通过以下迭代格式求解:

$$\begin{cases}X^{k+1}=arg\ \min\limits_{X}\{\ h_1(X,\mathcal{Z}_1^k)\}=arg\ \min\limits_{X}\ \{f(X,A^k,\mathcal{Y}^k)+\frac{\rho}{2}\parallel X-X^k\parallel_F^2\},\\ A^{k+1}=arg\ \min\limits_{A}\{\ h_2(A,\mathcal{Z}_2^k)\}=arg\ \min\limits_{A}\ \{f(X^{k+1},A^k,\mathcal{Y}^k)+\frac{\rho}{2}\parallel A-A^k\parallel_F^2\},\\ \mathcal{Y}^{k+1}=arg\ \min\limits_{\mathcal{Y}}\{\ h_3(\mathcal{Y},\mathcal{Z}_3^k)\}=arg\ \min\limits_{\mathcal{Y}}\ \{f(X^{k+1},A^{k+1},\mathcal{Y}^k)+\frac{\rho}{2}\parallel\mathcal{Y}-\mathcal{Y}^k\parallel_F^2\}.\end{cases}\tag{19}$$

注意到X 和A 的子问题可以再分解为三个独立的子问题, 而且$\mathcal{Y}$ 有显示解. 具体如下:

X子问题:

$$X_n^{k+1}=\begin{cases}(\alpha_n(A_n^k)^TA_n^k+\rho I_1)^{\dagger}(\alpha_n(A_n^k)^TY_{(n)}^k+\rho X_n^k),\ n=1,2,\\ arg\ \underset{X_3}{min}\{\frac{\alpha_3}{2}\parallel Y_{(3)}^k-A_3^kX_3\parallel_F^2+\lambda\parallel \Xi\mathcal{S}_\gamma(WX_3^T)\parallel_1+\frac{\rho}{2}\parallel X_3-X_3^k\parallel_F^2\},\ n=3.\end{cases}\tag{20}$$

A 子问题:

$$A_n^{k+1}=(\alpha_nY_{(n)}^k(X_n^{k+1})^T+\rho A_n^k)(\alpha_n(X_n^k)^TX_n^k+\rho I_2)^{\dagger},\ n=1,2,3.\tag{21}$$

$\mathcal{Y}$ 子问题:

$$\mathcal{Y}^{k+1}=\mathcal{P}_{\Omega^c}(\sum_{n=1}^{3}\alpha_nfold_n(\frac{\alpha_nA_n^{k+1}X_n^{k+1}+\rho Y_{(n)}^k}{\alpha_n+\rho}))+\mathcal{F}.\tag{22}$$

接下来, 我们来分析求解X_3 的细节. 特别地, X_3^{k+1} 等价于求解如下问题的解:

$$\underset{X}{min}\ \frac{\mu}{2}\parallel Y_{(3)}^k-A_3^kX_3\parallel_F^2+\parallel \Xi\mathcal{S}_\gamma(WX_3^T)\parallel_1+\frac{\rho_1}{2}\parallel X_3-X_3^k\parallel_F^2.\tag{23}$$

显然, 若$f(X_3):=\frac{\mu}{2}\|Y_{(3)}^k-A_3^kX_3\|_F^2+\frac{\rho}{2}\|X_3-X_3^k\|_F^2$ 和$g(WX_3^T):=\|\Xi\mathcal{S}_\gamma(WX_3^T)\|_1$, 优化问题(23) 是问题(9) 的一个特例. 因此, 可用临近不动点算法式(14)或式(15)进行求解. 不失一般性, 我们采用式(14), 则可得如下迭代格式:

$$\begin{cases}X_3^{k+1}=prox_{\lambda f}(X_3^k-\gamma W^TU^k),\\ U^{k+1}=prox_{\beta g^*}(U^k+\beta W(2X_3^{k+1}-X_3^k)).\end{cases}$$

进一步可得,

$$\begin{cases}X_3^{k+1}=\left((1+\lambda\rho_1)I+\lambda\mu(A_3^k)^TA_3^k\right)^{\dagger}\left(X_3^k-\lambda W^TU^k-\lambda\mu(A_3^k)^TY_{(3)}^k+\lambda\rho_1X_3^k\right),\\ U^{k+1}=prox_{\beta g^*}(U^k+\beta W(2X_3^{k+1}-X_3^k)).\end{cases}$$

对于给定的非负数ξ 和ϵ, 对任意$x\in\mathbb{R}$, 定义函数

$$l_{\xi,\epsilon}(x):=\xi\cdot max\{|\ x\ |-\epsilon,0\}.$$

则根据函数g的定义, 对任意矩阵$U=(u_{ij})$, 可知$g(U)=\sum_i\sum_j l_{\xi_{ij},\epsilon_{ij}}(u_{ij})$ 是可分离的. 因此, 其共轭函数满足

$$g^*(Y)=\sum_i\sum_j l^*_{\xi_{ij},\epsilon_{ij}}(y_{ij}),$$

且由临近算子性质(式(12)), 有

$$prox_{\beta g^*}(Y)=\left(prox_{\beta l^*_{\xi_{ij},\gamma_{ij}}}(y_{ij})\right).$$

最后,βg^* 的函数表达式与邻近算子可由如下引理求得:

引理 3.1 对于两个非负数ξ 和ϵ, 在$x\in\mathbb{R}$, 有

$$l^*_{\xi,\epsilon}(x):=\begin{cases}\epsilon\ |\ x\ |, & |\ x\ |\leq\xi;\\ +\infty, & 其它.\end{cases}$$

此外, 对$\beta > 0$, 都有

$$prox_{\beta l^*_{\xi,\epsilon}} := \begin{cases} sign(x)\xi, & |x| > \beta\epsilon + \xi; \\ x - sign(x)\beta\epsilon, & \beta\epsilon \leq \beta\epsilon + \xi; \\ 0, & \text{其它}. \end{cases} \tag{24}$$

引理3.1的详细证明见文献[28].

综合上述分析, 则得到了本文提出的求解模型(17)的算法:

算法 1 基于BSUM算法的AS-LRTC模型求解

输入: 观测张量:$\mathcal{F}$; 观测元素已知位置的集合:Ω; 给定的Tucker 秩:$r = (r_1, r_2, r_3)$.

输出: 重构张量:$\mathcal{Y}$.

```
function AS-LRTC
    初始化: A_n^0 = rand(I_n × r_n), X_n^0 = rand(r_n × ∏_{m=1,m≠n}^{k-1} I_m), n = (1,2,3); Y = P_Ω(F);
    while not converged do
        通过式(20)更新X_1, X_2;
        通过求解(23)更新X_3;
        通过式(21)更新A_n(n = 1,2,3);
        通过式(22)更新Y;
    end While
return Y
```

3.3 收敛性分析

接下来, 说明本文基于BSUM提出的算法满足引理2.1的条件, 从而必是收敛的.

定理 3.1 由(18) 生成的迭代格式收敛到坐标极小点:

证明: 首先, $h(\mathcal{Z}, \mathcal{Z}^k)$ 是函数$f(\mathcal{Z})$ 的近似, 是其在第k 次迭代的全局上界, 并满足下列条件:

$$\begin{aligned}
&h_i(\mathcal{Z}_i, \mathcal{Z}) = f(\mathcal{Z}), \forall \mathcal{Z}, i = 1,2,3; \\
&h_i(\bar{\mathcal{Z}}_i, \mathcal{Z}) \geq f(\mathcal{Z}_1, \cdots, \bar{\mathcal{Z}}_i, \cdots, \mathcal{Z}_3), \forall \bar{\mathcal{Z}}_i, \forall \mathcal{Z}, i = 1,2,3; \\
&h_1^{'}(\bar{\mathcal{Z}}_1, \mathcal{Z}; \mathcal{D}_1)|_{\bar{\mathcal{Z}}_1 = \mathcal{Z}_1} = f^{'}(\mathcal{Z}; \mathcal{D}^1), \forall \mathcal{D}^1 = (\mathcal{D}_1, 0, 0); \\
&h_2^{'}(\bar{\mathcal{Z}}_2, \mathcal{Z}; \mathcal{D}_2)|_{\bar{\mathcal{Z}}_2 = \mathcal{Z}_2} = f^{'}(\mathcal{Z}; \mathcal{D}^2), \forall \mathcal{D}^2 = (0, \mathcal{D}_2, 0); \\
&h_3^{'}(\bar{\mathcal{Z}}_3, \mathcal{Z}; \mathcal{D}_3)|_{\bar{\mathcal{Z}}_3 = \mathcal{Z}_3} = f^{'}(\mathcal{Z}; \mathcal{D}^3), \forall \mathcal{D}^3 = (0, 0, \mathcal{D}_3); \\
&h_i(\bar{\mathcal{Z}}_i, \mathcal{Z}) \text{在点} (\bar{\mathcal{Z}}_i, \mathcal{Z}) \text{ 处是连续的}, i = 1,2,3.
\end{aligned}$$

其中, $\mathcal{Z} = (X, A, \mathcal{Y})$, 当$i = 1,2,3$ 时$\mathcal{Z}_i$ 分别等于$X, A, \mathcal{Y}$.

另外, 子问题$h_i(i = 1,2,3)$ 分别关于$X, A, \mathcal{Y}$ 强凸, 因此每个子问题均有唯一解. 故按坐标收敛到问题的极小值点. □

4 数值实验

在本节中, 我们将在3阶张量数据上测试AS-LRTC方法的性能, 数据类型包括：视频数据、高光谱图像、核磁共振图像以及彩色图像. 进行对比的方法为Ji等人[26]提出的MF-TV和Jiang等人[27]提出的MF-Framelet 的张量填充方法. 为了定量评价各方法恢复的图像质量, 我们采用峰值信噪比(Peak Signal-to-Noise Ratio, PSNR)和结构相似性指数(Structural Similarity, SSIM)作为数值指标. 假设真实张量是$\bar{\mathcal{Y}}$, 则恢复出的张量$\mathcal{Y}$的PSNR的定义如下：

$$\mathrm{PSNR}(\mathcal{Y},\bar{\mathcal{Y}})=10log_{10}\frac{n\bar{\mathcal{Y}}_{max}^2}{\parallel\bar{\mathcal{Y}}-\mathcal{Y}\parallel_F^2}$$

其中n表示张量中的元素个数, $\bar{\mathcal{Y}}_{max}$表示真实张量中的像素最大值. 假设真实二维图像是$\bar{Y}$, 则恢复出的图像Y的SSIM定义如下：

$$\mathrm{SSIM}(Y,\bar{Y})=\frac{(2mean(Y)mean(\bar{Y})+c_1)(2cov(Y,\bar{Y})+c_2)}{(mean(Y)^2+mean(\bar{Y})^2+c_1)(std(Y)^2+std(\bar{Y})^2+c_2)}$$

其中$mean(Y,\bar{Y}),std(Y)$和$cov(Y,\bar{Y})$分别表示Y的像素平均值, Y的标准差, Y和$\bar{Y}$ 的协方差, c_1和c_2 为两个常数. 对于真实张量数据, 我们计算其按各维度展开矩阵的SSIM值的平均值, 以此代替张量的SSIM值. 停止准则是两个连续重构张量的相对变化(RelCha), 定义如下：

$$RelCha=\frac{\parallel Y^{k+1}-Y^k\parallel_F}{\parallel Y^k\parallel_F}<\varepsilon$$

其中$\varepsilon>0$是给定容许误差.

在所有实验过程中, 对于MF-TV, MF-Framelet方法, 其参数设置采用文献给出的参数. 本文提出的AS-LRTC模型参数设置如下: $\alpha_i=1/3(i=1,2,3)$, $\rho=0.1$, $\mu=1/3,\lambda=10,\beta=0.9/\lambda,\rho_1=0.01$. 所有实验均在Windows 10和MATLAB(R2016b) 平台上实现, 平台配置为2.70GHz的Intel(R)Xeon(R) CPU和32G的内存.

4.1 视频图像修复

在本节中, 我们测试了7个是视频: salesman, coastguard, news, suzie, hall, carphone, foreman. 所有视频均为YUV格式. 在实验中, 首先将视频的Y通道转化为150帧的灰度视频, 然后进行测试. 所有测试视频的尺寸均为$144\times176\times150$, 采样率SR分别设置为$5\%,10\%,20\%,30\%$和40%.

表1总结了通过三种低秩张量填充方法针对不同的SR重建的所有测试视频的PSNR和SSIM. 结果表明, 该方法在PSNR和SSIM值方面始终优于比较方法, 特别是在采样率SR 低于20% 时. 图4显示了在$SR=10\%$的情况下, 通过三种低秩张量填充方法重建的所有视频的其中一帧. 我们观察到所提出的AS-LRTC方法所重建视频的视觉效果优于其他方法. 具体而言, 所提出的AS-LRTC方法能够更好地填充缺失数据, 同时更多地保留视频的基本结构和有效信息, 而MF-TV和MF-Framelet 获得的结果仍然具有大量缺失数据. 图5中可以看出AS-LRTC方法恢复的视频的像素值能更好的接近原素值, 图6表明AS-LRTC方法恢复的每一帧图像的效果优于其它两种方法.

表 1: MF-TV, MF-Framelet和本文提出的AS-LRTC方法的数值对比结果(PSNR, SSIM)

video	SR	5%		10%		20%		30%		40%	
	method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
salesman	MF-TV	9.849	0.0634	13.882	0.2779	24.095	0.7981	32.108	0.9290	35.286	0.9640
	MF-Framelet	11.053	0.1652	14.634	0.4381	25.643	0.8334	32.623	0.9388	35.543	0.9669
	AS-LRTC	**18.2773**	**0.5465**	**27.592**	**0.8676**	**31.741**	**0.9298**	**34.202**	**0.9563**	**35.957**	**0.9701**
coastguard	MF-TV	7.532	0.0369	8.607	0.0615	11.463	0.1290	17.603	0.3529	26.503	0.7671
	MF-Framelet	8.945	0.0654	10.685	0.1154	14.589	0.2006	21.346	0.5072	28.180	0.8198
	AS-LRTC	**18.638**	**0.3150**	**19.405**	**0.4001**	**20.249**	**0.4922**	**23.727**	**0.7137**	**24.676**	**0.7703**
news	MF-TV	10.604	0.1179	12.638	0.2187	19.153	0.5277	30.356	0.8380	34.845	0.9302
	MF-Framelet	12.591	0.2575	15.062	0.3748	22.280	0.6472	32.730	0.904	35.820	0.9508
	AS-LRTC	**21.943**	**0.7306**	**26.155**	**0.8489**	**30.615**	**0.8986**	**33.921**	**0.9399**	**36.470**	**0.9626**
suzie	MF-TV	13.559	0.0915	22.126	0.6018	31.830	0.8713	34.808	0.9247	36.484	0.9481
	MF-Framelet	15.486	0.2014	24.040	0.6729	32.328	0.8810	35.009	0.9281	36.591	0.9489
	AS-LRTC	**20.503**	**0.5378**	**29.051**	**0.8042**	**33.178**	**0.8993**	**35.192**	**0.9315**	**36.606**	**0.9498**
hall	MF-TV	13.414	0.4184	24.779	0.8265	33.170	0.9412	35.025	0.9600	36.274	0.9690
	MF-Framelet	13.015	0.4671	24.118	0.8325	33.307	0.9450	**35.186**	**0.9619**	**36.423**	**0.9713**
	AS-LRTC	**26.655**	**0.8475**	**30.514**	**0.9110**	**33.654**	**0.9469**	35.173	0.9608	36.348	0.9695
carphone	MF-TV	10.058	0.1137	14.743	0.4782	30.762	0.8748	34.470	0.9390	36.208	0.9601
	MF-Framelet	10.640	0.2151	14.849	0.5344	31.651	0.8886	34.719	0.9447	36.363	0.9632
	AS-LRTC	**19.924**	**0.4798**	**29.611**	**0.8463**	**33.203**	**0.9235**	**35.073**	**0.9494**	**36.445**	**0.9636**
foreman	MF-TV	7.056	0.0253	13.749	0.2617	29.726	0.8471	32.780	0.9149	34.451	0.9421
	MF-Framelet	7.252	0.0277	13.505	0.3017	30.179	0.8561	32.891	0.9183	34.539	0.9441
	AS-LRTC	**15.503**	**0.1431**	**26.131**	**0.6978**	**31.126**	**0.8835**	**33.083**	**0.9231**	**34.583**	**0.9446**



图 4: 不同方法对视频图像其中一帧的恢复结果(采样率为10%). 第一列: 真实数据; 第二列: 观测数据; 第三列: MF-TV 恢复结果; 第四列: MF-Framelet 恢复结果; 第五列: AS-LRTC 恢复结果

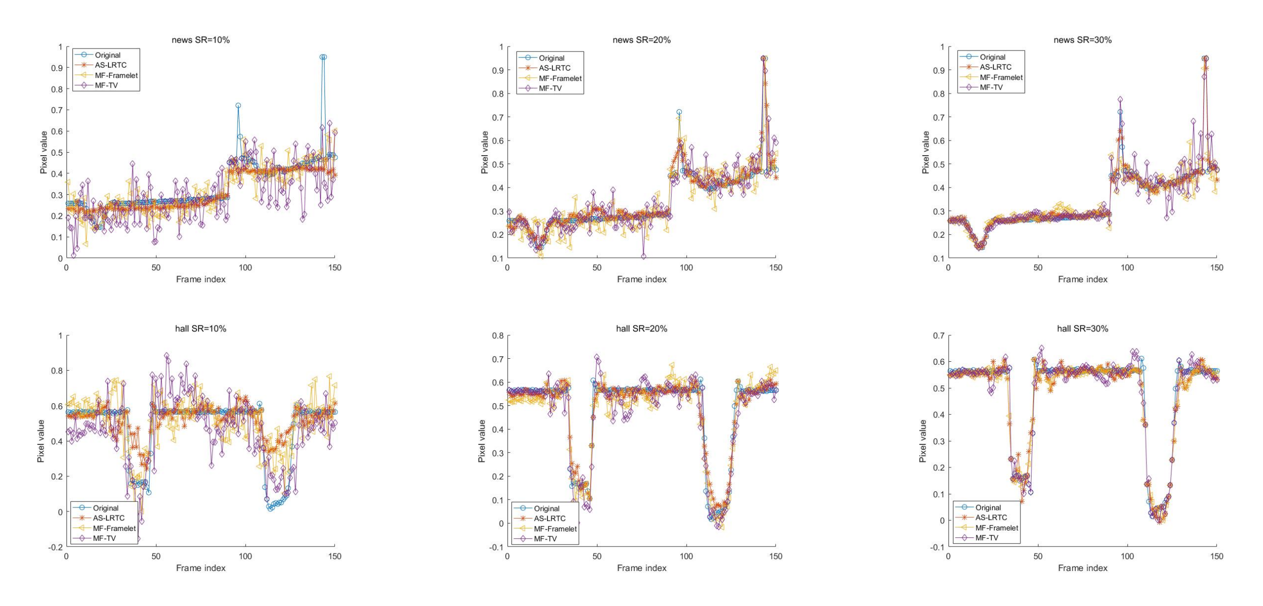


图 5: 对视频news和hall利用三种LRTC方法进行重建得到其中一个模-3纤维对应的像素值

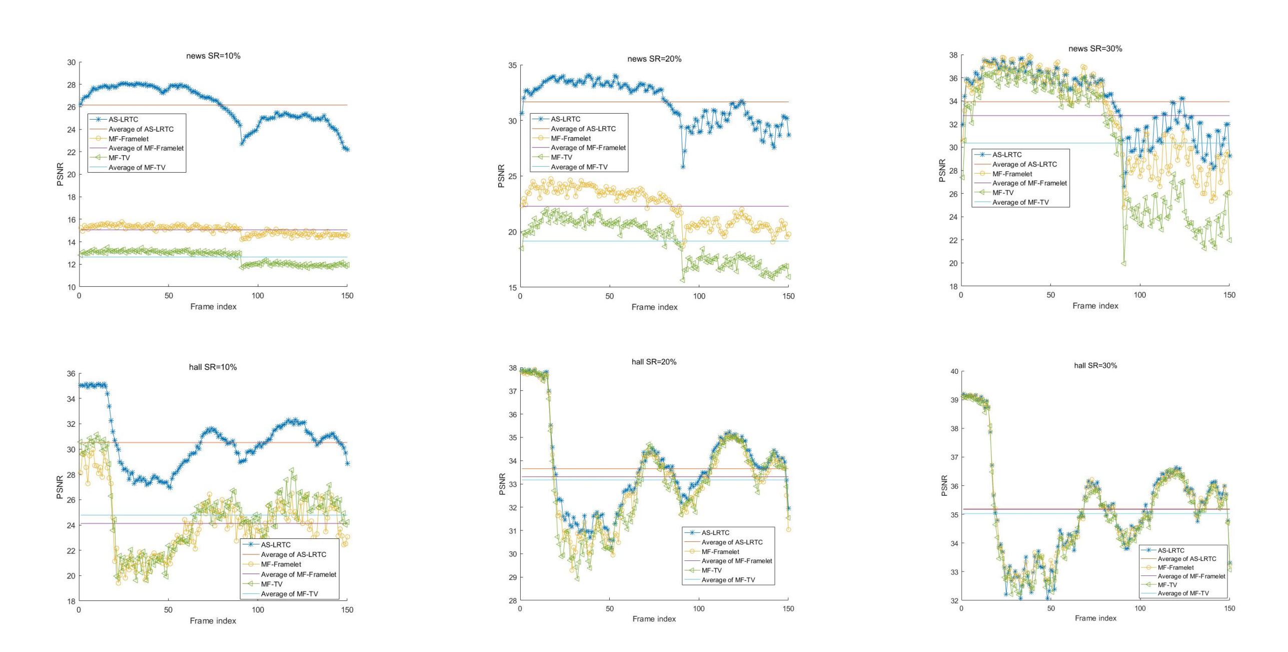


图 6: 对视频news和hall利用三种LRTC方法进行重建得到各帧的PSNR值

4.2 高光谱图像修复

在本节中, 我们测试的高光谱图像的大小为$350 \times 350 \times 188$. SR设置为10%. 图7 显示了通过提出的方法和三种比较方法重建的测试高光谱图像的波段. 可以看出, 与MF-TV和MF-framelet相比, 所提出的AS-LRTC方法能够很好的重构高光谱图像.

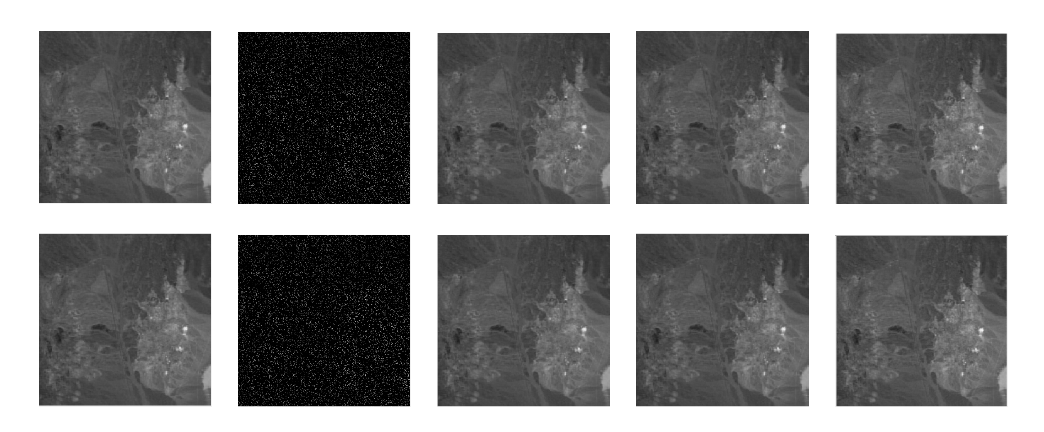


图 7: 不同方法对高光谱视频图像第56波段和第155 波段的恢复结果(采样率为10%). 第一列: 真实数据; 第二列: 观测数据; 第三列: MF-TV 恢复结果; 第四列: MF-Framelet 恢复结果; 第五列: AS-LRTC恢复结果

4.3 核磁共振图像修复

该测试使用尺寸为$181 \times 217 \times 181$ 的MRI数据作为测试数据. SR设置为10%. 由于可以将所有方向的切片视为图像, 因此在图8 中, 我们显示了从三个不同方向分别观察到的三个重建MRI 数据的代表性切片. 在不同方向的重构上, 我们看到, 本文所提出的AS-LRTC方法重构的MRI的视觉质量均优于其他的比较方法.

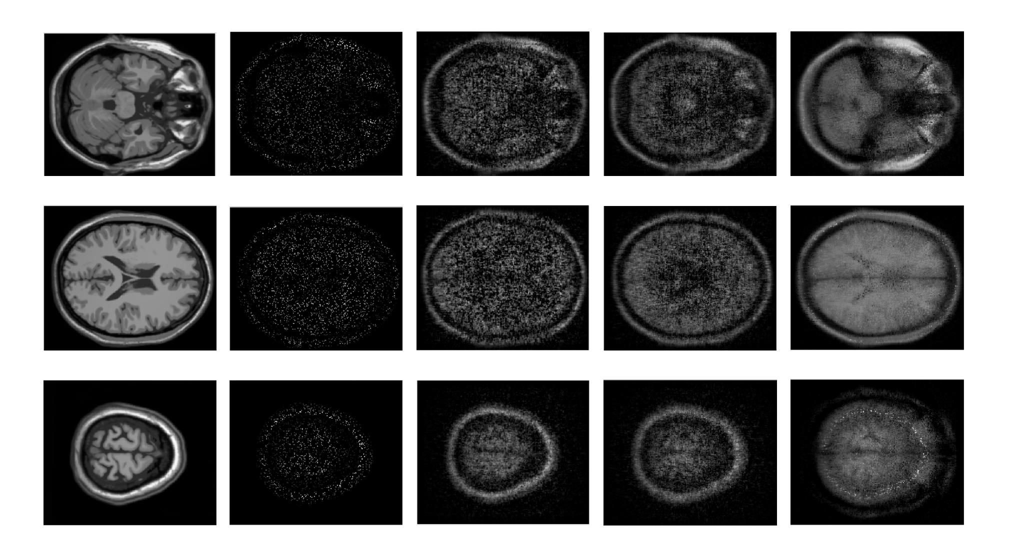


图 8: 不同方法对视频图像其中一帧的恢复结果(采样率为10%). 第一列: 真实数据; 第二列: 观测数据; 第三列: MF-TV 恢复结果; 第四列: MF-Framelet 恢复结果; 第五列: AS-LRTC 恢复结果

4.4 彩色图像修复

本节我们用$256 \times 256 \times 3$大小的彩色图片来测试我们的方法的有效性和优点, 根据彩色图片在空间域的分段光滑性, 我们将模型适当的改为(25), 从模型看, 求解算法与前文相似.

$$\begin{aligned} &\underset{\mathcal{Y},A,X}{min} \sum_{n=1}^{3} \frac{\alpha_n}{2} \parallel Y_{(n)} - A_n X_n \parallel_F^2 + \lambda \parallel \Xi \mathcal{S}_\gamma (W X_2^T) \parallel_1 \\ &s.t \ P_\Omega(\mathcal{Y}) = \mathcal{F}. \end{aligned} \tag{25}$$

从表2中可以看出, 我们的方法恢复出的彩色图片的PSNR 值和SSIM值比MF-TV 和MF-Framelet 方法高, 尤其是当采样率为10% 和20% 时, 我们的方法的效果更明显. 从图9中可以更直观的看出我们的方法恢复出的图像效果明显好于其它两种方法, 从背景的色彩上看, 我们的方法更接近原图像.

表 2: MF-TV, MF-Framelet和本文提出的AS-LRTC方法的数值对比结果(PSNR, SSIM)

name	SR	10%		20%		30%		40%		50%	
	method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
airplane	MF-TV	8.213	0.0390	15.110	0.1467	22.541	0.5633	27.702	0.7839	30.930	0.5407
	MF-Framelet	8.950	0.0441	15.725	0.1769	23.375	0.5966	28.082	0.7991	31.408	0.8831
	AS-LRTC	**18.038**	**0.4094**	**23.254**	**0.6018**	**26.707**	**0.7627**	**29.546**	**0.8528**	**31.760**	**0.8949**
facade	MF-TV	12.805	0.1776	20.365	0.5481	27.361	0.8407	31.072	0.9215	33.661	0.9541
	MF-Framelet	15.563	0.3127	22.459	0.6487	28.337	0.8685	31.619	0.9297	33.942	0.9573
	AS-LRTC	**19.195**	**0.4951**	**27.809**	**0.8561**	**30.558**	**0.9155**	**32.763**	**0.9462**	**34.411**	**0.9618**
house	MF-TV	10.375	0.0573	17.407	0.2468	25.2175	0.6329	30.249	0.8041	33.298	0.8806
	MF-Framelet	11.573	0.0750	19.095	0.3064	26.432	0.6704	30.672	0.8171	33.348	0.8844
	AS-LRTC	**19.904**	**0.4356**	**25.559**	**0.6493**	**29.210**	**0.7880**	**32.132**	**0.8649**	**34.092**	**0.9044**
lena	MF-TV	10.960	0.0779	14.664	0.1565	19.848	0.3452	25.607	0.6444	30.413	0.8356
	MF-Framelet	12.685	0.1122	16.367	0.1898	21.142	0.3999	26.396	0.6773	30.540	0.8406
	AS-LRTC	**16.426**	**0.3044**	**17.494**	**0.3648**	**26.189**	**0.7090**	**29.285**	**0.8281**	**31.940**	**0.8932**
peppers	MF-TV	10.771	0.0759	14.071	0.1553	17.675	0.2843	23.326	0.5633	28.557	0.7822
	MF-Framelet	12.172	0.1441	15.986	0.2101	18.907	0.3338	24.295	0.6049	29.202	0.7984
	AS-LRTC	**14.482**	**0.2270**	**17.235**	**0.3662**	**24.819**	**0.6606**	**28.021**	**0.7914**	**30.720**	**0.8668**
sailboat	MF-TV	9.306	0.0962	12.681	0.1894	15.839	0.2961	19.606	0.4563	24.886	0.6740
	MF-Framelet	11.237	0.1438	15.414	0.2566	18.132	0.3652	21.448	0.5235	25.736	0.7055
	AS-LRTC	**15.279**	**0.2791**	**16.304**	**0.3527**	**23.3674**	**0.6454**	**25.814**	**0.7533**	**28.415**	**0.8395**

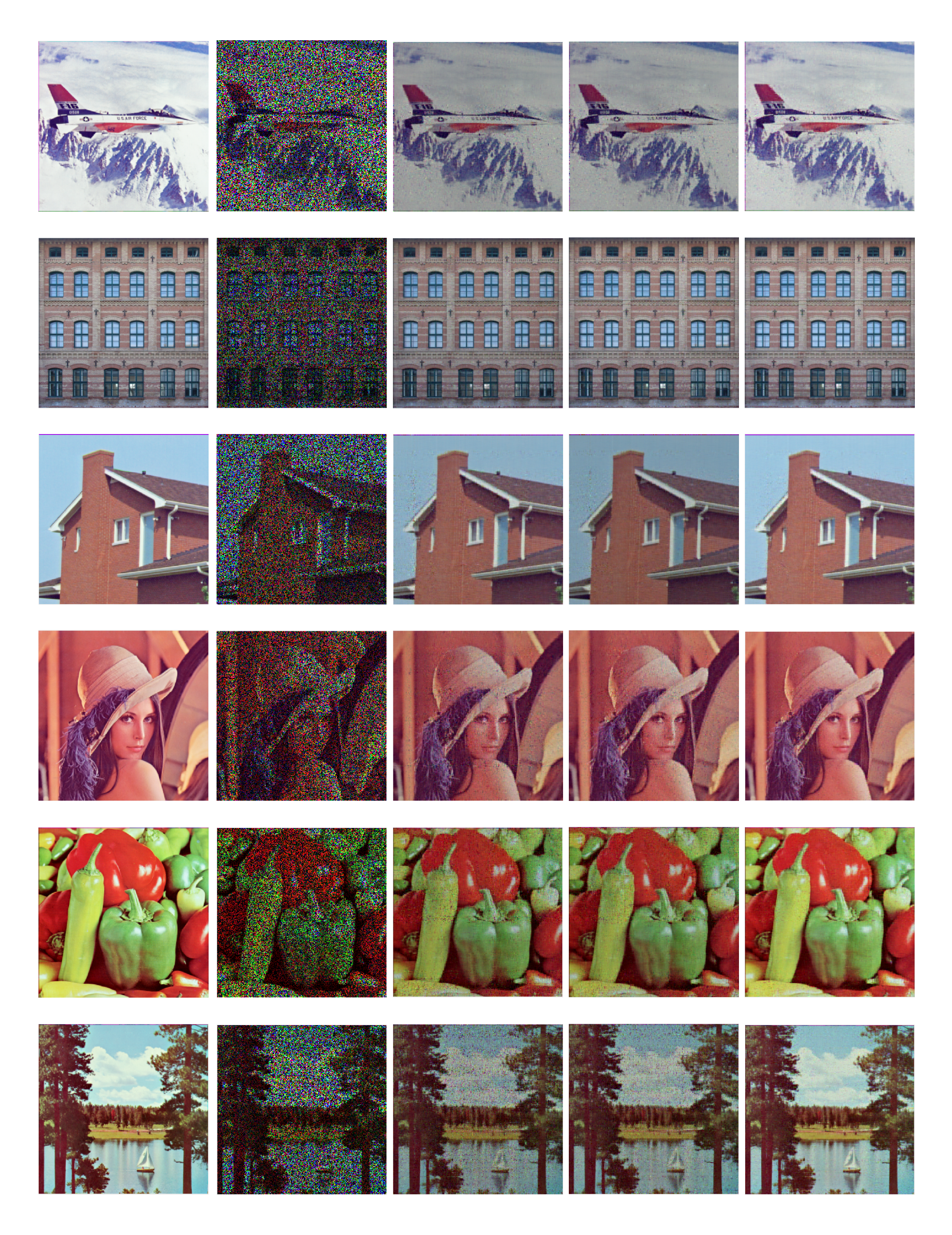


图 9: 不同方法对彩色图像的恢复结果(采样率为40%). 第一列: 真实数据; 第二列: 观测数据; 第三列: MF-TV 恢复结果; 第四列: MF-Framelet 恢复结果; 第五列: AS-LRTC 恢复结果

5 结论

本文结合近似稀疏和低秩矩阵分解提出了一个近似稀疏的张量填充模型, 利用问题的结构设计了有效的AS-LRTC算法. 一定条件下证明了算法的收敛性, 大量数值结果表明该方法在恢复视觉效果和增强

分段平滑度方面均取得了很好的效果.

参考文献

[1] Yokota T, Lee N, Cichocki A. Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria. IEEE Transactions on Signal Processing, 2016, 65(5): 1196-1206

[2] Mei J J, Dong Y, Huang T Z, et al. Cauchy noise removal by nonconvex ADMM with convergence guarantees. Journal of Scientific Computing, 2018, 74(2): 743-766

[3] Sobral A, Zahzah E. Matrix and tensor completion algorithms for background model initialization: A comparative evaluation. Pattern Recognition Letters, 2017, 96: 22-33

[4] Varghees V N, Manikandan M S, Gini R. Adaptive MRI image denoising using total-variation and local noise estimation//IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012). IEEE, 2012: 506-511

[5] Zhao X L, Wang F, Huang T Z, et al. Deblurring and sparse unmixing for hyperspectral images. IEEE Transactions on Geoscience and Remote Sensing, 2013, 51(7): 4045-4058

[6] Chen Y, Huang T Z, Zhao X L, et al. Hyperspectral image restoration using framelet-regularized low-rank nonnegative matrix factorization. Applied Mathematical Modelling, 2018, 63: 128-147

[7] Zhang H, Patel V M. Convolutional sparse and low-rank coding-based rain streak removal//2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017: 1259-1267

[8] Jiang T X, Huang T Z, Zhao X L, et al. Fastderain: A novel video rain streak removal method using directional gradient priors. IEEE Transactions on Image Processing, 2018, 28(4): 2089-2102

[9] Sidiropoulos N D, De Lathauwer L, Fu X, et al. Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing, 2017, 65(13): 3551-3582

[10] Liu J, Musialski P, Wonka P, et al. Tensor completion for estimating missing values in visual data. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(1): 208-220

[11] Tan H, Cheng B, Wang W, et al. Tensor completion via a multi-linear low-n-rank factorization model. Neurocomputing, 2014, 133: 161-169

[12] Filipovic M, Jukic A. Tucker factorization with missing data with application to low-nrank tensor completion. Multidim. Syst. Sign. P, 2013

[13] Hillar C J, Lim L H. Most tensor problems are NP-hard. Journal of the ACM (JACM), 2013, 60(6): 1-39

[14] Gandy S, Recht B, Yamada I. Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Problems, 2011, 27(2): 025010

[15] Romera-Paredes B, Pontil M. A new convex relaxation for tensor completion//Advances in Neural Information Processing Systems. 2013: 2967-2975

[16] Li Y F, Shang K, Huang Z H. Low Tucker rank tensor recovery via ADMM based on exact and inexact iteratively reweighted algorithms. Journal of Computational and Applied Mathematics, 2018, 331: 64-81

[17] Ji T Y, Huang T Z, Zhao X L, et al. A non-convex tensor rank approximation for tensor completion. Applied Mathematical Modelling, 2017, 48: 410-422

[18] Mu C, Huang B, Wright J, et al. Square deal: Lower bounds and improved relaxations for tensor recovery//International conference on machine learning. 2014: 73-81

[19] Nie J. Nearly Low Rank Tensors and Their Approximations. arXiv preprint arXiv:1412.7270, 2014

[20] Bai M, Zhang X, Ni G, et al. An adaptive correction approach for tensor completion. SIAM Journal on Imaging Sciences, 2016, 9(3): 1298-1323

[21] Xu Y, Hao R, Yin W, et al. Parallel matrix factorization for low-rank tensor completion. Inverse Problems & Imaging, 2015, 9 (2) : 601-624.

[22] Wen Z, Yin W, Zhang Y. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Mathematical Programming Computation, 2012, 4(4): 333-361

[23] Zhao X L, Wang W, Zeng T Y, et al. Total variation structured total least squares method for image restoration. SIAM Journal on Scientific Computing, 2013, 35(6): B1304-B1320

[24] Chan T F, Vese L A. Active contours without edges. IEEE Transactions on image processing, 2001, 10(2): 266-277

[25] Chan S H, Khoshabeh R, Gibson K B, et al. An augmented Lagrangian method for total variation video restoration. IEEE Transactions on Image Processing, 2011, 20(11): 3097-3111

[26] Ji T Y, Huang T Z, Zhao X L, et al. Tensor completion using total variation and low-rank matrix factorization. Information Sciences, 2016, 326: 243-257

[27] Jiang T X, Huang T Z, Zhao X L, et al. Matrix factorization for low-rank tensor completion using framelet prior. Information Sciences, 2018, 436: 403-417

[28] Shen L, Xu Y, Zhang N. An approximate sparsity model for inpainting. Applied and Computational Harmonic Analysis, 2014, 37(1): 171-184

[29] Kolda T G, Bader B W. Tensor decompositions and applications. SIAM review, 2009, 51(3): 455-500

[30] Razaviyayn M, Hong M, Luo Z Q. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM Journal on Optimization, 2013, 23(2): 1126-1153

[31] Ron A, Shen Z. Affine systems inL2 (Rd): the analysis of the analysis operator. Journal of Functional Analysis, 1997, 148(2): 408-447

[32] Cai J F, Chan R H, Shen Z. A framelet-based image inpainting algorithm. Applied and Computational Harmonic Analysis, 2008, 24(2): 131-149

[33] Chai A, Shen Z. Deconvolution: A wavelet frame approach. Numerische Mathematik, 2007, 106(4): 529-587

[34] Li Q, Micchelli C A, Shen L, et al. A proximity algorithm accelerated by Gauss - Seidel iterations for L1/TV denoising models. Inverse Problems, 2012, 28(9): 095003

[35] Li Q, Shen L, Xu Y, et al. Multi-step fixed-point proximity algorithms for solving a class of optimization problems arising from image processing. Advances in Computational Mathematics, 2015, 41(2): 387-422

[36] Zhang X, Burger M, Osher S. A unified primal-dual algorithm framework based on Bregman iteration. Journal of Scientific Computing, 2011, 46(1): 20-46